Peer-Review 2: UML Sequence Diagram

Alessandro Paolo Gianni Callegari, Leonardo Chiaretti, Matteo Civitillo, James Enrico Busato

Gruppo AM19

Valutazione del diagramma UML Sequence Diagram del gruppo AM28.

Il seguente diagramma presenta sette suddivisioni, ognuna delle quali gestisce ogni fase del gioco e comunicazione tra Client e Server in modo completo e ordinato.

# Lati positivi

# L'UML è ben diviso e organizzato a seconda delle diverse fasi di gioco in cui ci si trova, mostrando anche eventuali zone di loop in cui il server o client si ritrovano a reiterare l'azione; i nomi dei metodi sono chiari ed anche i parametri contenuti al loro interno, come le azioni che il server svolge una volta ricevuto un messaggio.

# Lati negativi

# Come consigliato dai professori e tutor del corso, non c'è bisogno che il server risponda sempre ad un messaggio del client, in quanto una comunicazione "simmetrica" costituisce un maggior tempo di latenza e di conseguenza un allungamento dei tempi di gioco; inoltre, la comunicazione deve sempre partire dal client, con il server che si limita a rispondere.

# Ad esempio, nel caso della richiesta del numero di giocatori, noi abbiamo ridotto lo scambio di messaggi Client-Server in modo tale che alla prima connessione di un client il server risponda con un messaggio di conferma e incorporando anche la richiesta per il numero di giocatori, migliorando i tempi.

# Anche nel caso della pescata di una carta, non c'è bisogno che il server restituisca un messaggio di successo, poiché la mossa viene riconosciuta come tale nel momento in cui viene aggiornata la View (ad esempio quando il client manda una richiesta del metodo draw() il server manda direttamente l'aggiornamento per la view).

# Confronto tra le architetture

Molto interessante la richiesta del ping presente nella quinta suddivisione del diagramma che non mancheremo di implementare anche noi, per il resto le nostre architetture si somigliano molto nella logica comunicativa.